DZero Monte Carlo Production

Ideas for CMS

Greg Graham
Fermilab CD/CMS
1/17/01

Hcal/Jets/Met Meeting

Monte Carlo Production
An Overview

Monte Carlo Production Systems

- Monte Carlo Verification I-Code
  - Patches, Versions, Executables
- RED : Runtime Environment Distribution
- Monte Carlo Verification II-Executables
  - Check representative output from all platforms
- Monte Carlo Specification
  - Physics Process (H), Request specification (H)
- Control and Tracking : The Grid
  - Condor pools, Globus
  - Special Tools

Monte Carlo Verification - I

- Before anything is distributed, make sure :
  - Correct versions of external code (eg - Pythia) are being used.
  - All needed patches have been applied
  - The output looks good at the build site

Runtime Environment Distribution

- D zero : the "mini-tarfiles"
  - Offline production executables from official builds
  - Packaged with needed .so and databases
  - Installed on top of each other in regular way so as not to destroy previous installations
- It is not optimal to distribute the build environments and compile everywhere
  - missing patches, different static libs, (?) compilers, etc.
  - $\sigma_{MC\text{-}site}$? : systematic error from site to site MC variation - minimize this, but make it measurable

Monte Carlo Verification II

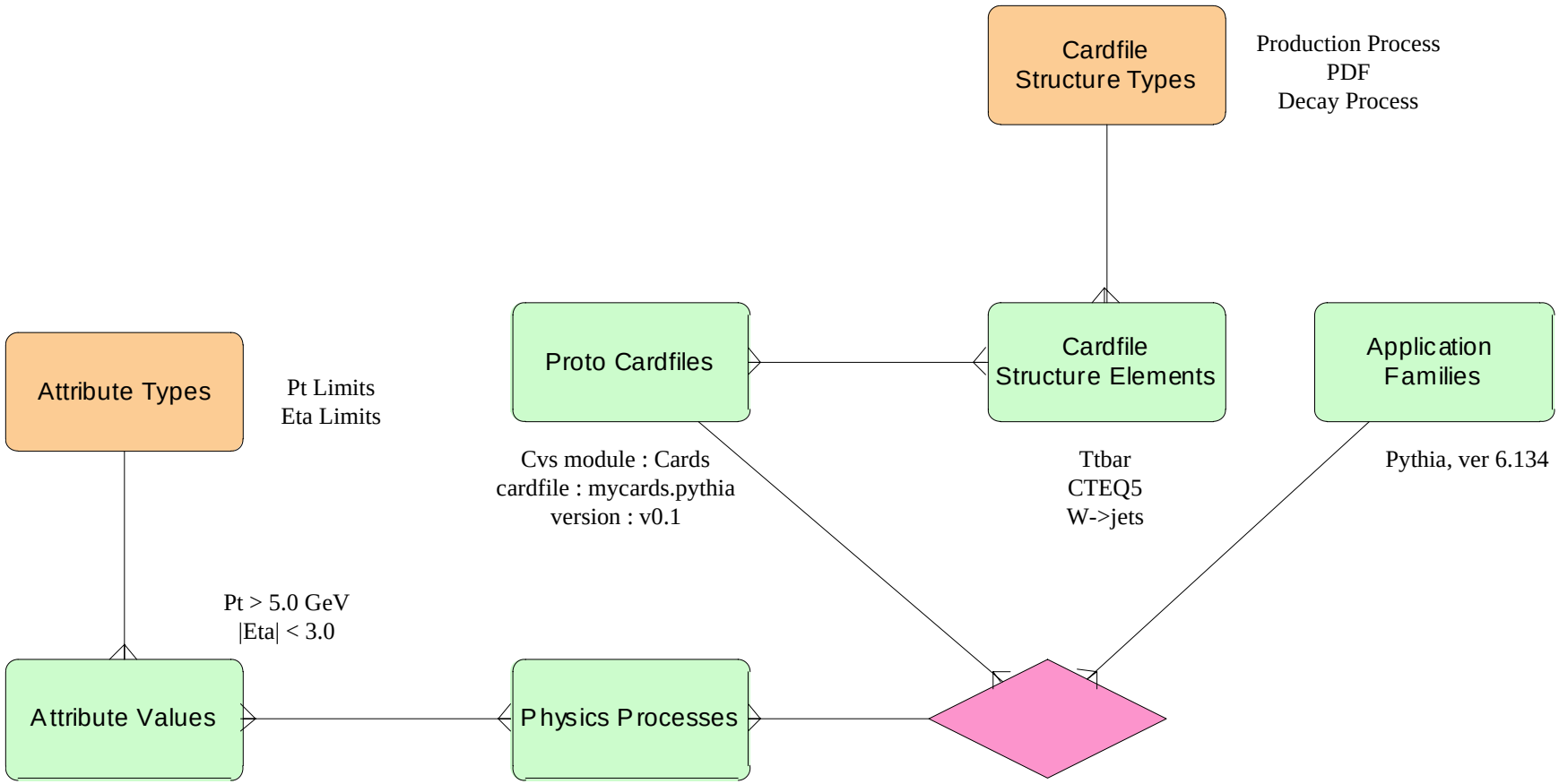
- After executables have been distributed, make sure :
  - standard samples are run at different sites
  - output is compared from site to site

Physics Process Specification

This was harder than it seemed at the outset.

- Schema developed for relational DB system
  - Keeps track of cardfiles
    - Cardfiles required to be stored in CVS
  - Keeps track of cardfile elements
    - production, decay, PDF, etc.
  - Attributes (eg Pt limit, Eta limit) and cardfile elements can be added dynamically
  - Attributes plus cardfiles define what we mean by Physics Process

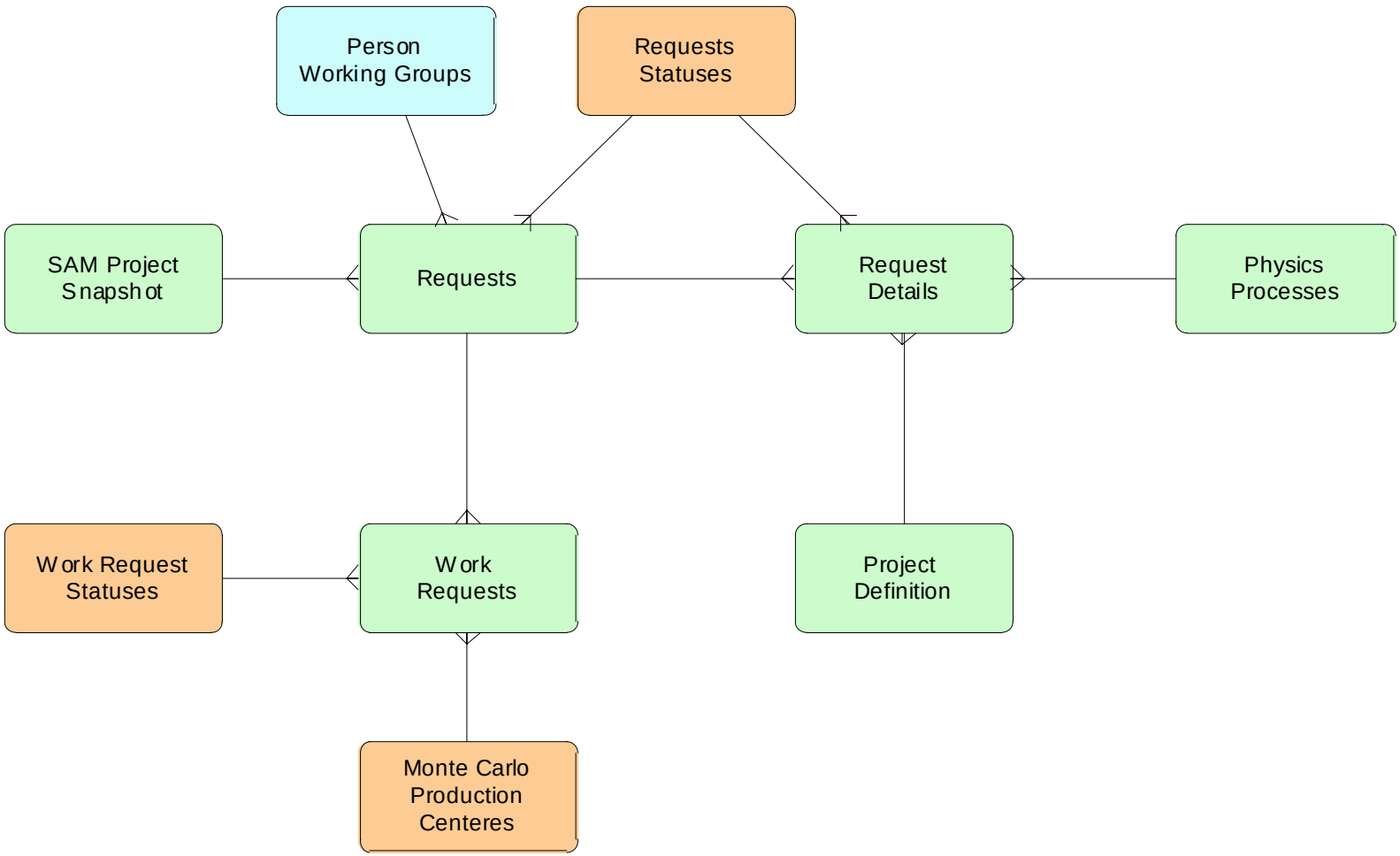
Physics Process Specification



Request Specification

- Schema developed for Relational DB system
  - Keeps track of processing requests
    - List can grow dynamically; keeps track of whole processing chain
  - Keeps track of groups of input/output files
    - In D zero, SAM "projects" are defined on the input/output files
  - Requests can be split into site-specific "work-requests"
  - In D zero, appearance of files with the correct parentage in SAM constitutes job tracking

Automatic Request Processing Schema



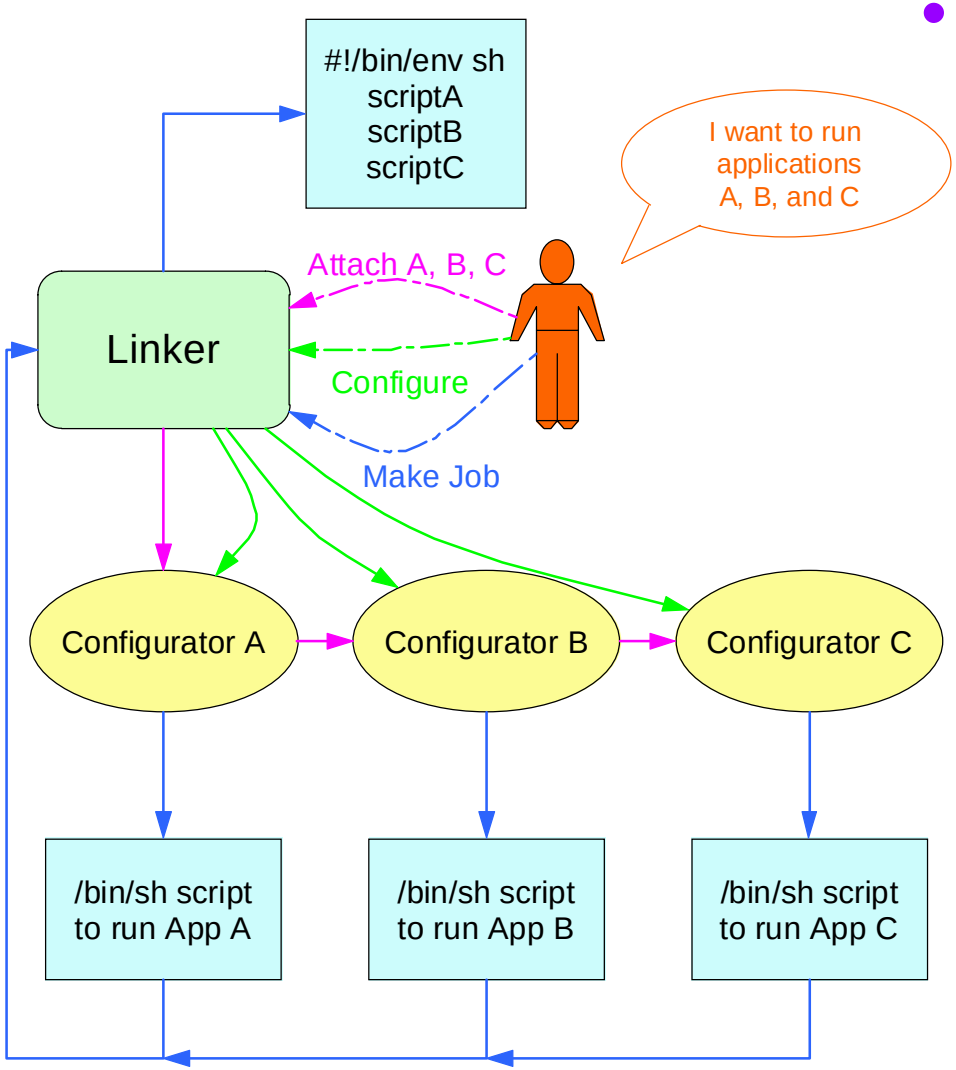
MonteCarlo Job Specification

- mc_runjob : A D0 package written in Python
  - Driven by macro scripts
    - Macro scripts serve to specify processing steps in a regular way
  - Runs most D0 Offline Executables
    - All in principle …
  - Chains executables together
    - Run generation through ntuple in one step
    - Handles naming of input and output files
    - Handles seed generation, run numbers, etc.

The Grid

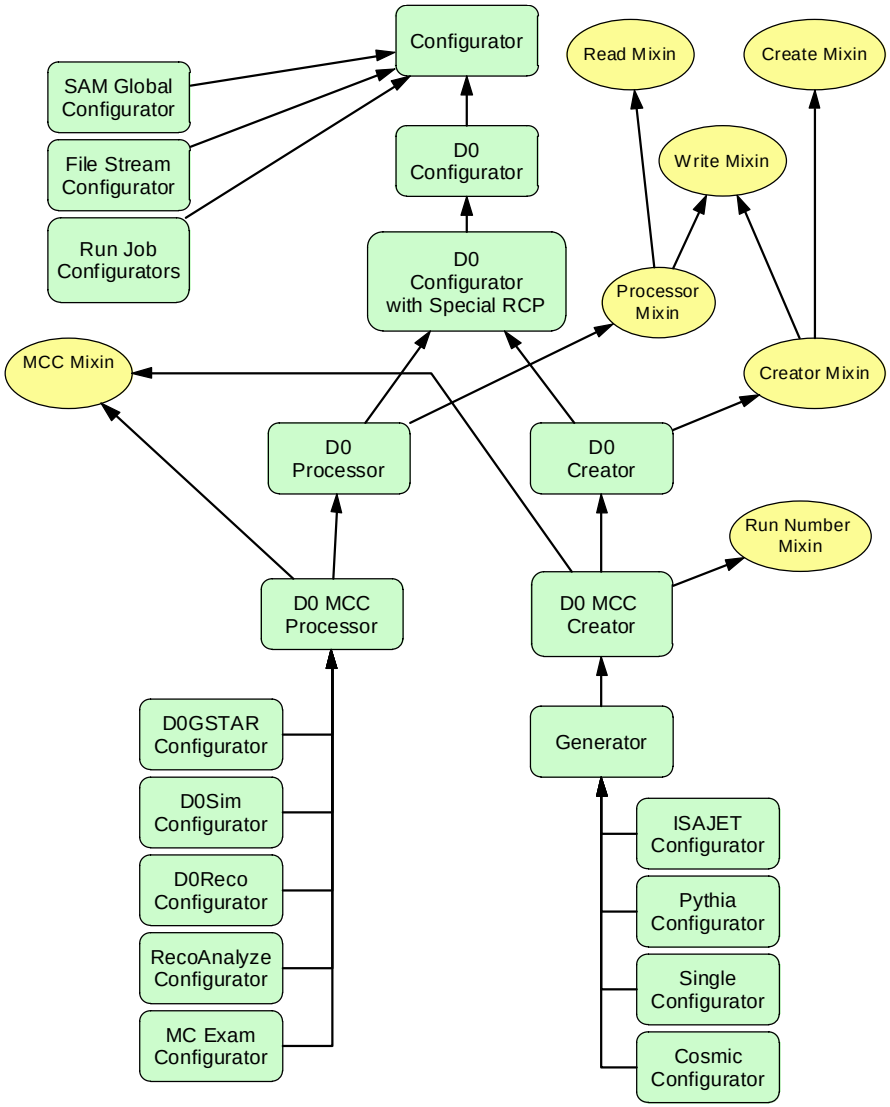
- Many experimental projects are underway for the benefit of CMS Production
  - EuroGrid, PPDG, GriPhyN
  - Existing Tools: Globus, Condor
  - Hardware: Farms & Clusters, Huge SMPs, …
- The Vision: How can we go beyond production and bring the power of the hardware to individual physicists?
  - Dzero: 100K events/day = 200 events/day/physicist; CMS may certainly have more capacity.
  - Do we dare let anyone run their own MC like in the days of Fortran?

Specific Dzero Tools



- mc_runjob
  - "Configurators"
    - Run and configure each DØ offline executable
  - The "Linker"
    - Links Configurators
    - Collects shellscripts that actually do the work
  - Macro scripts
    - The Linker and Configurators obey a regular set of macro commands in macro scripts

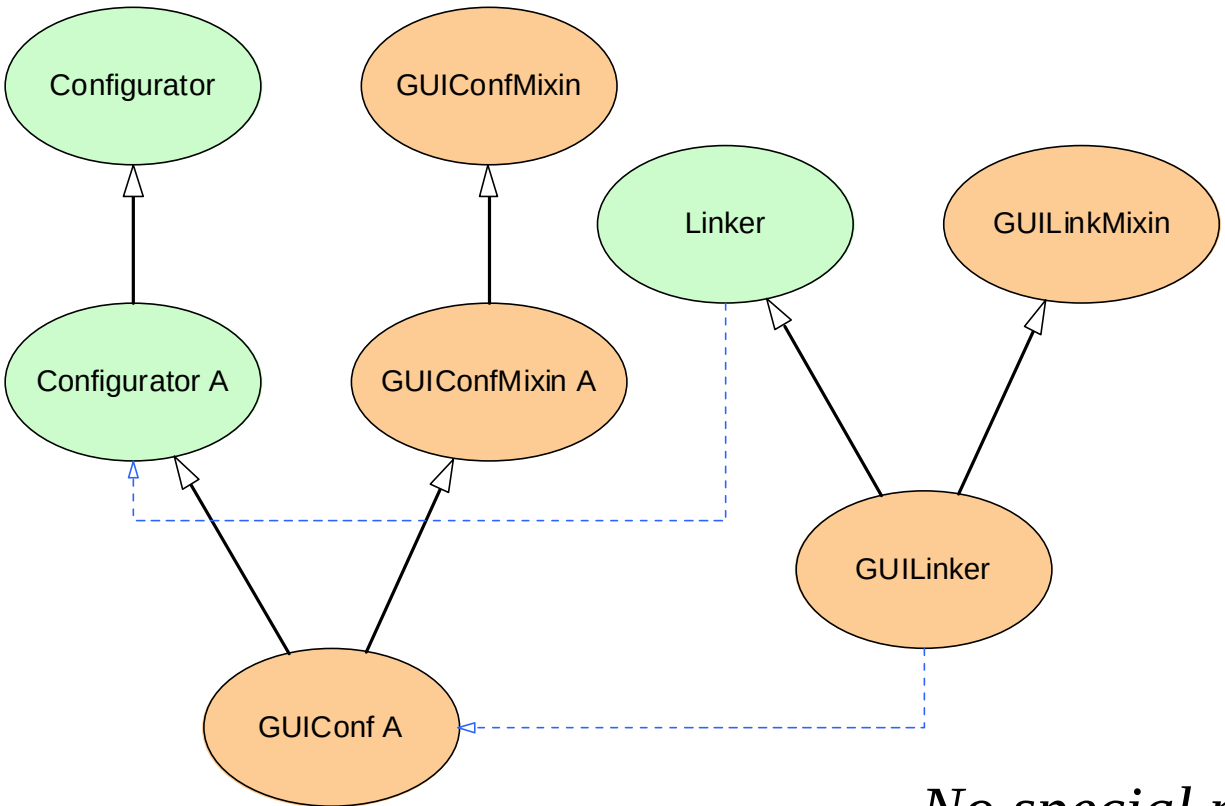
Can easily be extended !



- Inheritance Hierarchy
  - Inheritance is used to incorporate D0 specific configuration (ie -RCP, runtime environment, naming, SAM metadata)
  - Other special purpose configurators can handle the batch system, manage a list of files, etc.

Can easily be extended !

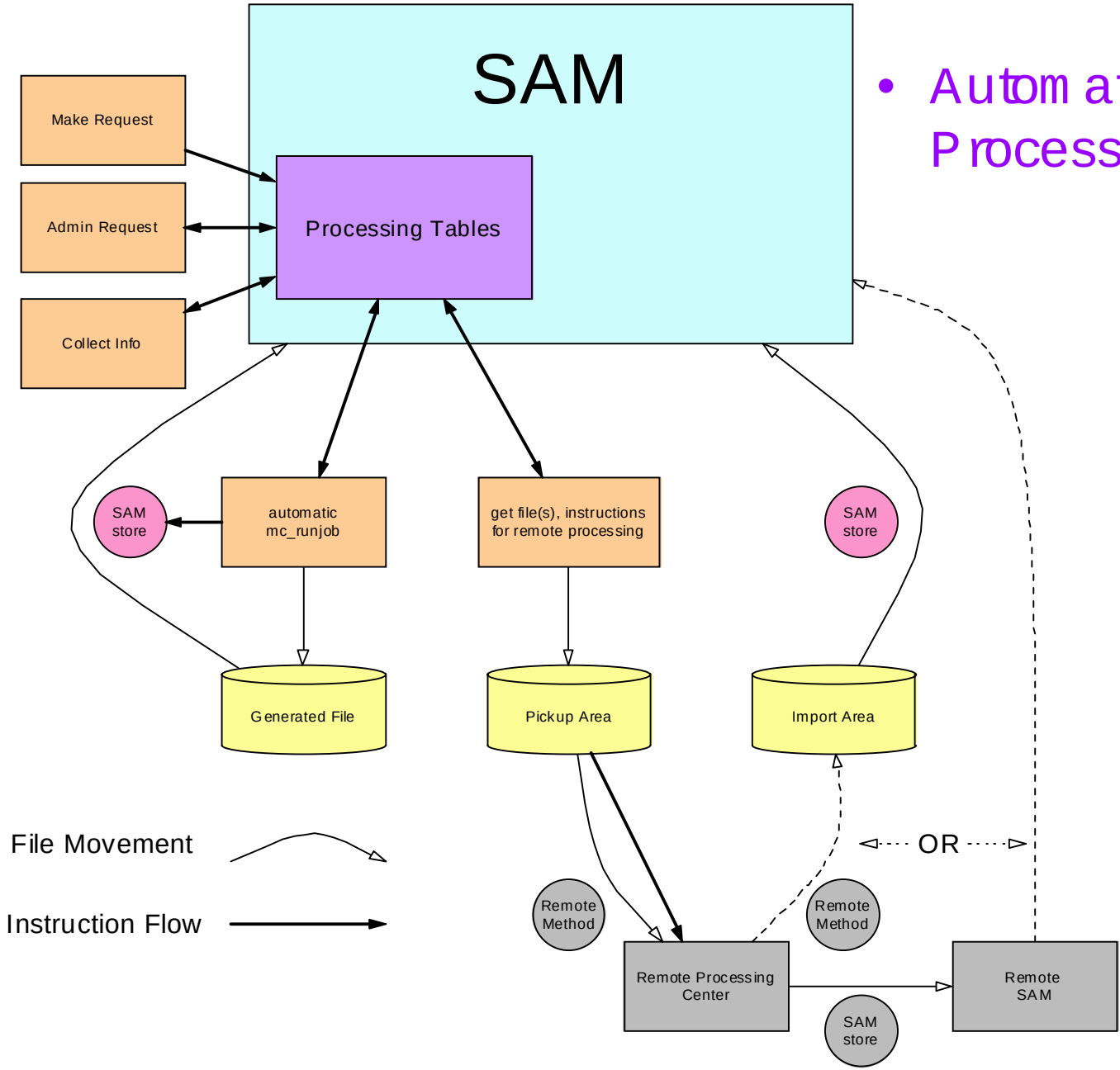
GraphicalSupport



- While primarily a macro script driven tool, the mc_runjob inheritance hierarchy also provides core tools with GUI methods that build a GUI from information contained in the configurators themselves.

No special porting task for GUI !

- Automatic Request Processing System



Remote Processing Centers

- Monte Carlo simulation, digitization, and eventually some MC reconstruction will be handled offsite.
- Currently operating RPCs
  - UTArlington; IN2P3, Lyon; NIKHEF, Amsterdam; Prague
- Future planned RPCs
  - Lancaster; Rio de Janeiro; TIFR
- Generated events will be stored with SAM
- We expect better than 100K events/day

References

- G. E. Graham "The D zero Monte Carlo Challenge" Proceedings, CHEP 2000
- G. E. Graham "D zero Monte Carlo" Proceedings, ACAT 2000 (to appear soon)